Dlaždičkové programování
aneb informatikova lázeň

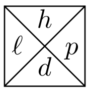
(verze 1.0 z 2021-04-01)*

0. Dlaždičky v koupelně

Také občas přemýšlíte nad matematickými úlohami v koupelně? My v ní občas i programujeme. Stačí se trochu zasnít nad úžasnými vzory barev na dlaždičkovaných stěnách … a najednou je z nich kompletní počítač! Ten budeme dnes zkoumat.

Pojďme si nejprve říct, co je to taková dlaždice. *Dlaždice* představuje čtverec jednotkové velikosti, který má každou hranu obarvenou nějakou *barvou.* Obarvení jednotlivých hran může a nemusí být stejné, ale každá hrana musí být obarvena právě jednou barvou. Dohodněme se také, že barvy budeme označovat nějakými symboly, typicky čísly a písmeny. Barev bude vždy konečně mnoho.

Často budeme pro názornost dlaždice zobrazovat opravdu jako čtverce rozdělené na čtyři části, ale formálně dlaždici zavedeme jako uspořádanou čtveřici (ℓ, h, p, d), kde jednotlivé symboly ℓ, h, p, d označují po řadě obarvení levé, horní, pravé a dolní hrany dlaždice. Kreslit ji budeme takto:



Z takových dlaždic můžeme skládat *dláždění.* Prostoru, který chceme dlaždičkovat, říkejme *zeď.* Ta je obdélníková a má rozměry $r \times s$ (přičemž jednotkou bude délka hrany dlaždice). Okraje zdi jsou rozděleny na úseky o jednotkové délce a každý z těchto úseků má podobně jako hrany dlaždic nějaké obarvení.

Jako dláždění označujeme pokrytí zdi dlaždicemi, pokud toto pokrytí splňuje několik podmínek. V první řadě požadujeme, aby v každém z $r \times s$ čtverců byla umístěna právě jedna dlaždice. Dále každé dvě sousední dlaždice musí mít ty hrany, kterými se dotýkají, obarvené stejnou barvou. Požadavek na stejnou barvu máme i na okrajové dlaždice, tedy dlaždice, které přiléhají k okraji zdi, musí mít příslušnou hranu obarvenou stejnou barvou, jakou je obarvený příslušný úsek okraje. Poslední podmínka: dlaždice nesmíme při tvorbě dláždění otáčet. Dodejme ještě, že každou dlaždici smíme použít libovolně-krát.

* Vychází z textu, který napsala Karry Burešová pro KSP.

Pomocí dláždění, resp. jeho existence nebo neexistence, můžeme snadno rozhodovat úlohy, na které se odpovídá ANO, nebo NE. Jak to uděláme? Musíme sestavit vhodnou množinu dlaždic, ze kterých budeme smět vybírat při tvorbě dláždění. Horní okraj zdi obarvíme podle vstupu. Ještě potřebujeme obarvit ostatní okraje, s tím, že všechny jejich úseky obarvíme stejnou barvou (můžeme o tom tedy uvažovat jako o obarvení celého okraje jednou barvou). Dlaždice a obarvení vybíráme tak, aby dláždění existovalo, právě když odpověď na úlohu je ANO.

Možných dláždění (a jim příslušných množin dlaždic a obarvení okrajů) může existovat velmi mnoho, tak si je alespoň trochu omezme. Požadujeme, aby zeď byla vždy široká právě tak, jak dlouhý je vstup. Horní okraj tedy bude vstupu přesně odpovídat. Navíc chceme, aby výška zdi byla nejmenší možná.

To, co jsme před chvilkou popsali, je *dlaždicový program.* Ten se skládá z nějaké konečné množiny dlaždic a nějakého obarvení okrajů zdi. Formálně by se jednalo o uspořádanou čtveřici (D, ℓ, p, d), kde D je množina dlaždic a ℓ, p, d představují obarvení levého, pravého a dolního okraje.

Program na zadaný vstup odpoví ANO, pokud je možné vydláždit nějakou zeď dlaždicemi z množiny D tak, aby horní okraj byl obarven podle vstupu a zbývající okraje barvami ℓ, p a d. Neexistuje-li žádné takové dláždění, výstupem programu je NE.

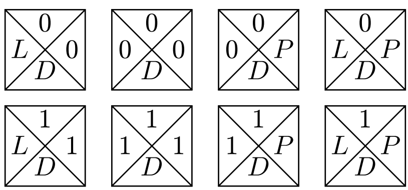
Dlaždicové programy jsou vzácná zvířátka, a tak jim budeme na vstupu předkládat pouze neprázdné řetězce.

Bývá hezké umět u programů ve výpočetním modelu určovat složitost. U dlaždicových programů to zvládneme jednoduše: za dobu výpočtu prohlásíme minimální výšku zdi, pro kterou existuje dláždění (časová složitost je pak maximum z dob výpočtu přes všechny vstupy dané délky), použitou paměť pak představuje plocha vydlážděné zdi. Vstupy, na něž je odpověď NE, takže žádné dláždění neexistuje, složitost neovlivní.

Dost bylo teoretizování, pojďme se podívat na konkrétní příklad.

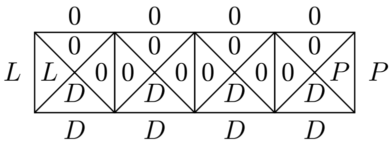
1. Ošplíchnutí na začátek

Mějme na vstupu nějakou posloupnost nul a jedniček. Naším úkolem je rozhodnout, jestli je tato posloupnost konstantní, tedy zda obsahuje pouze nuly nebo pouze jedničky. Využijeme k tomu dlaždicový program s následujícími dlaždicemi. Můžeme je rozdělit do čtyř typů, každý typ existuje ve dvou barvách:



Levý okraj obarvíme L, pravý P, dolní D. Náš program určitě odpoví správně, a dokonce mu k tomu bude stačit jeden řádek. Takové odvážné tvrzení by se ale slušelo dokázat.

Jelikož všechny dlaždičky mají dolní hranu obarvenou D a žádná nemá barvou D obarvenou hranu horní, buď bude mít dláždění výšku 1, nebo vůbec nepůjde vytvořit. Pro konstantní posloupnost jistě dláždění existuje. V případě jednoprvkové posloupnosti použijeme příslušnou dlaždici čtvrtého typu, v případě posloupnosti delší pomocí vhodné dlaždice prvního typu „zvolíme číslo“ a následně ho „propagujeme“ až k pravému okraji:



Platí i to, že vše, pro co dláždění existuje, musí být konstantní posloupnost. K levému i k pravému okraji může přiléhat vždy jen jedna konkrétní dlaždice (podle hodnoty na vstupu), a k jejich spojení je potřeba „předávat“ stále stejné číslo.

V následujících úkolech bude vaším cílem sestrojit dlaždicový program řešící danou úlohu. Nutnou součástí je důkaz, že program opravdu úlohu řeší – tedy že odpovídá jak ANO*, tak* NE *právě tehdy, když má.*

Úkol 1.1: Vyřešte negaci našeho příkladu: Dostanete posloupnost nul a jedniček. Odpovězte ANO, právě když tato posloupnost *není* konstantní.

Úkol 1.2: Dostanete posloupnost jedniček. Zjistěte, zda tato posloupnost má sudou délku.

Úkol 1.3: Dostanete posloupnost nul a jedniček. Zjistěte, zda je v ní počet jedniček dělitelný třemi.

Úkol 1.4: Dostanete posloupnost čísel z množiny $\{0, \dots, 9\}$. Je tato posloupnost neklesající?

Úkol 1.5: Napište program, který má stejnou barvu levého a pravého okraje a řeší následující úlohu: Je dána řada jedniček. Je její délka násobkem 3 nebo 5?

Úkol 1.6: Najděte úlohu, která se dá řešit dlaždicovým programem, ale nikoliv takovým, který má stejnou barvu levého a pravého okraje.

Úkol 1.7: Ukažte, jak pro každé $k \geq 1$ navrhnout dlaždicový program, který dostane posloupnost nul a jedniček a zjistí, zda se jedná o dvojkový zápis čísla dělitelného k.

2. Bublinková koupel

Úkol 2.1: Dostanete posloupnost nul a jedniček. Odpovězte ANO, pokud je nul stejný počet jako jedniček.

Úkol 2.2: Dokažte, že vaše řešení předchozí úlohy je nejlepší možné. Tedy že každý dlaždicový program, který tuto úlohu řeší, spotřebuje (asymptoticky) aspoň tolik času jako váš program. Kdybyste si nevěděli rady, zkuste alespoň dokázat, že konstantní čas nestačí.

Úkol 2.3: Dostanete posloupnost levých a pravých závorek. Zjistěte, zda je správně uzávorkovaná.

Úkol 2.4: Dostanete posloupnost písmen `a` a `b`. Odpovězte ANO, pokud je tato posloupnost palindromická – tedy stejná odpředu jako odzadu.

Úkol 2.5: Dokažte, že vaše řešení předchozí úlohy je (asymptoticky) nejlepší možné.

3. Vypouštíme vanu aneb programy, co mají výstup

Uvažujme na chvíli, že bychom místo ANO nebo NE odpovídali obecným řetězcem. I to se dá ve světě dlaždicových programů popsat – místo aby dolní okraj zdi byl obarven zadanou barvou, využijeme barvy jednotlivých dlaždic na dolním okraji k zapsání výstupu.

Dlaždicový program nám pak bude odpovídat na otázku „je y správným výstupem pro vstup x?“, což je zase otázka na ANO/NE. Budeme přitom vyžadovat, aby pro každý vstup x program přijal jenom jeden výstup y. Můžeme omezit podmnožinu barev, které se smí vyskytovat ve vstupech a výstupech, takze „pracovní“ barvičky používané programem během výpočtu se nebudou s výstupem plést.

Úkol 3.1: Na vstupu dostaneme dvě čísla ve dvojkové soustavě (můžete si zvolit, jak přesně budou zapsaná). Spočítejte jejich součet.

Úkol 3.2: Na vstupu dostaneme posloupnost znaků z množiny $\{\mathtt{a}, \ldots, \mathtt{z}\}$. Setřiďte ji (uspořádejte znaky podle abecedy).

Úkol 3.3: Na vstupu dostaneme posloupnost písmen `a` a `b` sudé délky. Prohoďte první a druhou polovinu.

4. Heuréka! aneb trocha meta-otázek a jiné balneofilosofie

Už jsme vymysleli programy, které řeší různé zapeklité úlohy. Pojďme teď přemýšlet obecněji nad tím, jaké úlohy se dají dlaždicovými programy řešit a jak rychle.

Úkol 4.1: Mějme problém, který se dá vyřešit dlaždicovým programem pracujícím v čase t, kde t je nějaká konstanta. Dokažte, že existuje jiný dlaždicový program, který odpovídá na tutéž otázku, ale stačí mu čas 1.

Úkol 4.2: Vymyslete, jak se dlaždicové programy dají simulovat na klasickém počítači. Ukažte, že pro každý dlaždicový program existuje klasický program, který spočítá totéž. Omezte se pro začátek na „jednořádkové“ dlaždicové programy (běžící v čase 1). Snažte se dosáhnout co nejlepší složitosti klasického programu vzhledem k délce vstupu; závislost na počtu dlaždic dlaždicového programu nás nezajímá.

Úkol 4.3: Představte si, že máte speciální dlaždičkový program, který ale krom popisu dlaždiček a barvy stěn obsahuje navíc popis speciální dlaždičky, která musí být v každém dláždění použita nejvýše jednou. Musí nutně existovat „klasický“ dlaždičkový program, který počítá totéž? Co kdybychom speciální dlaždici museli použít nikoli nejvýše jednou, ale právě jednou?

Úkol 4.4: O vlastnosti $\varphi(x)$ řetězců nul a jedniček řekneme, že je *rozpoznatelná,* pokud existuje dlaždicový program, který odpovídá ANO právě tehdy, když vstup splňuje φ. Dokažte, že pokud φ a ψ jsou rozpoznatelné vlastnosti, $\varphi \wedge \psi$ a $\varphi \vee \psi$ jsou také rozpoznatelné.

Úkol 4.5: Dokažte, že pokud je φ rozpoznatelná programem, který běží v konstantním čase, pak $\neg\varphi$ je také rozpoznatelná.

Úkol 4.6: Platí obecně, že je-li φ rozpoznatelná, pak $\neg\varphi$ je také?